

Co-autoria Distribuída de Cursos na Internet utilizando CORBA e Java

Fernando A. M. Trinta, Carlos A. G. Ferraz

E-mail: {famt, cagf}@cin.ufpe.br

Universidade Federal de Pernambuco

Centro de Informática

Cx. Postal 7851, CEP 50.732-970, Recife – PE – Brasil

Fone: (0xx81) 271-8430, Fax: (0xx81) 271-8438

RESUMO

Atualmente, aplicações da Internet para fins de Educação a Distância constituem um dos campos mais atrativos para pesquisadores e educadores. Este artigo apresenta arquiteturas alternativas para suporte à co-autoria de aulas através da Internet. Tais arquiteturas foram modeladas considerando a distribuição de informações e serviços, com soluções baseadas no paradigma de objetos distribuídos, em particular, utilizando a especificação CORBA. Ao final, é apresentado um protótipo, implementado em Java, da arquitetura considerada mais apropriada.

Palavras-chave: Educação a Distância, Co-autoria, Objetos Distribuídos, CORBA, Java.

ABSTRACT

Nowadays, Internet-based Distance Learning applications constitute one of the most attractive fields for researches and educators. This paper presents alternative architectures for supporting Internet-based co-authoring of classes. Such architectures have been modeled considering the distribution of information and services, with solutions based on the distributed objects paradigm, in particular, using the CORBA specification. In the end, a prototype of the most appropriate architecture, implemented in Java, is presented.

Keywords: Distance Learning, Co-authoring, Distributed Objects, CORBA, Java.

1 Introdução

Nos últimos tempos, o surgimento e a proliferação das redes de computadores, e principalmente da Internet, têm criado um novo palco para experimentos de Educação a Distância (EAD), sendo o mesmo apontado como uma verdadeira revolução para a prática do ensino-aprendizagem. Aplicações de EAD, por serem naturalmente distribuídas, devem aproveitar da estrutura oferecida pelas redes de computadores, principalmente a grande abrangência como a Internet.

A Educação a Distância através da Internet apresenta uma série de vantagens em relação a tradicionais formas de EAD até então conhecidas como cursos por correspondência ou cursos através de redes de rádio e televisão. A principal vantagem está relacionada à interação entre as partes envolvidas no processo educacional. Através da Internet, alunos e professores podem comunicar-se de forma mais fácil, rápida e eficaz [2]. Tal comunicação pode ser tanto de forma assíncrona, através de e-mail e/ou grupos de notícias, ou de forma síncrona, através de *chats* de texto convencionais ou multimídia, permitindo que haja com isso comunicação em tempo real entre aluno e professor. Além desta vantagem, surge a possibilidade através da

Internet, que se criem grupos de alunos para aprendizado cooperativo, com tais grupos formados por alunos possivelmente separados geograficamente.

Há vários experimentos de utilização da Internet para EAD. Um exemplo simples de uma efetiva prática de utilização da Internet para fins de Educação a Distância é o uso da infraestrutura da *web* para divulgação de material didático através de páginas HTML (*Hypertext Markup Language*), comunicação de forma assíncrona através de *e-mail*, e uso de *newsgroups* para esclarecimento de dúvidas [10] [9]. Porém, tal forma de utilização subutiliza o potencial que a Internet pode oferecer para a Educação a Distância. A real utilização da Internet para fins educacionais deve ser realizada através de ambientes, sistemas desenvolvidos por instituições acadêmicas ou empresas privadas, onde são divulgados cursos, realizadas aulas e interações entre professores e alunos. Atualmente existe um bom número destes sistemas e ambientes que se dispõem a realizar EAD através da Internet, como por exemplo o AulanetTM da PUC-Rio [1] e o Projeto Virtus [20] da Universidade Federal de Pernambuco (UFPE). Através destes ambientes, alunos de vários pontos da rede podem freqüentar ou mesmo oferecer cursos das mais diversas áreas do conhecimento.

Embora tais ambientes tenham obtido sucesso na utilização da Internet para EAD, muito ainda há para se evoluir. A maior parte dos ambientes e sistemas que utilizam a Internet como meio para EAD, estão baseados num modelo centralizado de *Website*, com material didático exposto na forma de páginas HTML e comunicação assíncrona entre aluno e professor [10]. No entanto, a visão mais adequada para Educação a Distância utilizando a Internet seria a de “comunidades virtuais”, onde através das quais, grupos de professores cooperam entre si para produção de cursos, materiais didáticos, etc. para grupos de alunos [4] [5]. Além disso, espera-se que sejam utilizados recursos diversos para elaboração do material didático disponibilizado para os alunos.

Este artigo se preocupa com um aspecto em particular da visão supracitada: a co-autoria de aulas e a distribuição de informações e serviços em um ambiente de EAD. Aqui são mostradas arquiteturas que possibilitam a vários professores a co-autoria (construção de material didático de forma cooperativa) de aulas através da Internet [4] utilizando o paradigma de objetos distribuídos [9] [24]. Como resultado de tais arquiteturas, um protótipo foi implementado utilizando a linguagem Java [22] para o desenvolvimento da aplicação e CORBA [16] para a comunicação entre os objetos. O trabalho contribui para enfatizar a necessidade de um suporte a professores na preparação de material didático em ambientes educacionais *on-line*, apresentando uma ferramenta de co-autoria cooperativa, composta de objetos distribuídos para concordar com a natureza inerentemente distribuída de aplicações de EAD.

2 Co-autoria Distribuída de Aulas

Como explicado anteriormente, a Educação a Distância na Internet tem como uma de suas principais vantagens em relação a outras formas de ensino a distância, a aprendizagem cooperativa, onde através da qual estudantes reúnem-se em grupos, ajudando uns aos outros no processo de aprendizagem, atuando como parceiros entre si, e também com os professores, com o objetivo de adquirir conhecimento sobre um dado domínio [20]. Na Internet, esta idéia é semelhante, com a diferença que os integrantes do grupo estão fisicamente separados, distribuídos geograficamente.

De fato, a idéia trabalho cooperativo é mais vinculada a grupos de alunos, onde tarefas eram divididas entre os integrantes de um grupo, e no final o trabalho de todos era reunido, formando com isso uma solução mais completa a um problema. Porém, com a Internet, há a possibilidade que o mesmo conceito possa ser estendido também a grupos de professores,

permitindo que os mesmos possam interagir em grupos para a construção, e até apresentação de aulas e cursos à distância para grupos de alunos.

Essa possibilidade de cooperação pode ser bastante proveitosa para os professores, visando uma troca de experiências e abordagens educacionais, possibilitando a produção de materiais de aula com conteúdo ainda mais rico, além de se apresentar como uma excelente oportunidade para a reciclagem entre professores. Embora, tal possibilidade apresente tais benefícios, são poucos os ambientes desenvolvidos atualmente de EAD na Internet, que apresentam algum tipo de ferramenta que permita tal interação entre professores.

Na maioria dos atuais ambientes de EAD na Internet, ocorre até uma certa colaboração entre autores de material didático, onde, por exemplo, um professor associa *links* ao seu *site* para outros *sites* que contêm material que servirão de apoio a suas aulas, objetivando alcançar um conteúdo mais abrangente para seu próprio material. Porém, neste caso, não existe uma real colaboração entre os professores [9] .

O que se pensa como real cooperação entre autores de material para ensino a distância se daria na forma de discussões, troca de informações, construção simultânea de material didático e apresentação de aulas aos alunos por parte de um ou mais autores. A idéia passada reforça a utilização do termo “comunidade virtual”, onde comunidades virtuais de professores interagem com comunidades virtuais de alunos.

A questão da co-autoria vai além da criação do próprio material do curso apenas por professores. Através da Internet, alunos podem ter acesso a informações relativas às aulas e cursos que vão além daquelas que lhe são passadas nos mesmos. Muitas dessas informações são bastante relevantes, e seria de grande valia que as mesmas fossem incorporadas ao conteúdo dos cursos, de modo que todos os alunos tenham acesso a estas novas informações, promovendo com isso uma constante renovação dos materiais dos cursos. Através de suas contribuições, o aluno pode se sentir estimulado a participar de um ambiente que se utilize desta característica de autoria. A idéia é permitir aos alunos, a produção de seus próprios materiais de curso. Para tanto, faz-se necessário que todos os recursos que são disponibilizados a instrutores para criação de cursos também possam ser utilizados por alunos, com ou sem distinção ou restrições.

Além destas possibilidades, o conteúdo das aulas dos cursos também é um fator que merece atenção para autores. Atualmente, os recursos utilizados por autores de cursos na Internet baseiam-se em páginas HTML. Outros autores disponibilizam seu material em forma de documentos vinculados a programas específicos, como o Microsoft® Word ou Microsoft® PowerPoint. A maior parte do material divulgado nas formas citadas anteriormente são baseadas em texto, gráficos e imagens estáticas. A utilização de mídias contínuas, como vídeo e áudio, é quase inexistente. Deve ser permitido ao autor de um curso inserir em suas aulas, trechos de áudio e vídeo, pois tais recursos facilitam em muito, o aprendizado.

2.1 Co-autoria e objetos distribuídos

Os requisitos apresentados para um sistema de co-autoria de aulas com tais características envolvem um grau de complexidade bastante alto e um processamento distribuído de informações, que levaram os autores a buscar soluções através do paradigma de objetos distribuídos. Esta abordagem se faz necessária pela própria natureza distribuída que a Internet possui, associada à atual falta de suporte apropriado para a criação de aplicações distribuídas na rede.

Através desta abordagem, o ambiente para realização de co-autoria deve ser formado por vários objetos, que devem compor aplicações que processam dados espalhados em vários servidores na rede. Tais servidores guardam as diversas informações que podem ser colocadas nos slides da aula, como vídeos de assuntos correlatos às aulas, transparências ao estilo

PowerPoint, páginas HTML ou quaisquer outros recursos que possam ser levados aos alunos por meio dos professores às aulas. Como exemplo, imagine o cenário descrito pela figura 1.

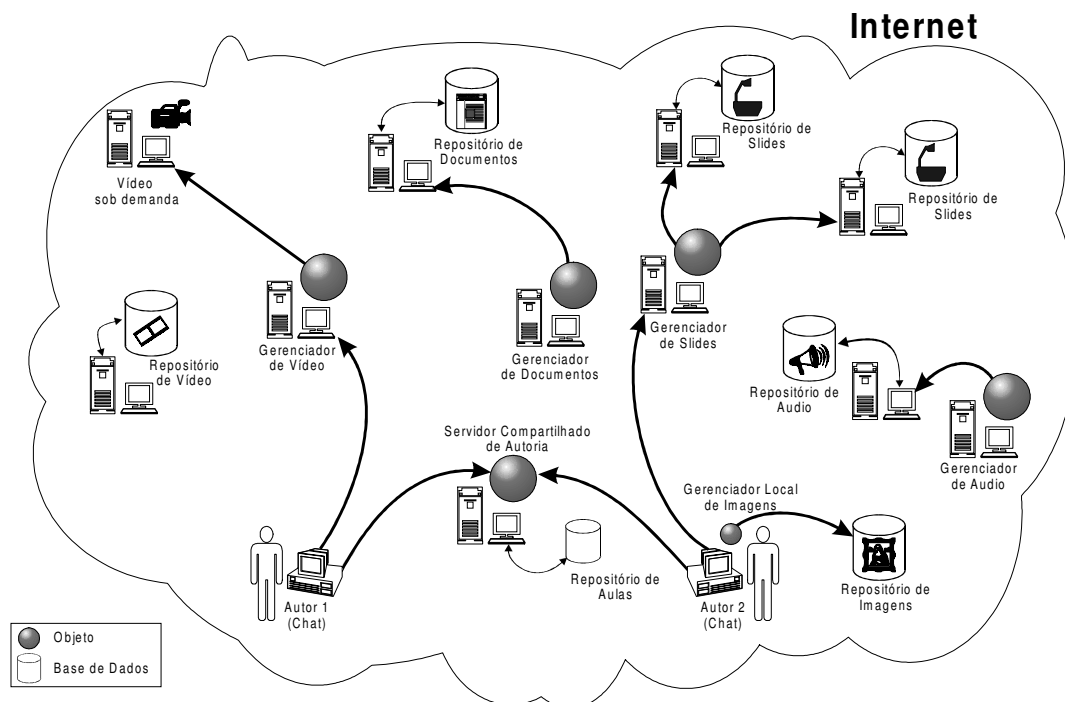


Figura 1 – Co-autoria distribuída de aulas na Internet.

A figura 1 apresenta um possível cenário onde usuários, chamados então de autores pela sua função de criação das aulas, interagem através de objetos distribuídos. Os autores utilizam um objeto que serve como servidor compartilhado para autoria das aulas, e que fornece suporte para que os autores montem as aulas, e utilizem os recursos desejados por cada um dos participantes. Como exemplo, na mesma figura temos dois autores, que interagem por meio de *chat*, mas que poderiam se comunicar por áudio ou videoconferência, de acordo com os recursos disponíveis (largura de banda da rede, por exemplo).

Os recursos por sua vez, encontram-se espalhados pela Internet, como servidores de vídeos, repositórios de imagens, áudio, slides e demais recursos. Objetos distribuídos são utilizados como gerenciadores para cada um destes recursos, onde os mesmos podem encontrar-se local ou remotamente.

Na figura 1, mostra-se que o autor 1 utiliza o gerenciador de vídeo para incorporar a possibilidade de vídeo sob demanda na aula. O autor 2, por sua vez, utiliza um gerenciador de slides para ter acesso a slides previamente definidos, armazenados em bases de dados remotas. Ressalta-se que com a utilização de objetos distribuídos, estes recursos podem estar replicados, possibilitando com isso uma melhor utilização dos mesmos e melhoria de desempenho, através do balanceamento de utilização dos servidores por parte dos usuários.

O autor 2 também tem acesso a uma base de dados de imagens, que pode ser acessada através de um gerenciador de imagens local a este autor, ressaltando-se que há transparência em relação a localização dos objetos utilizados, ou seja, o objeto cliente utilizado por um autor se conecta a objetos servidores independentemente de suas localizações na rede – transparência de localização, entre outras, é suportada por plataformas de distribuição, como CORBA.

O servidor compartilhado de autoria tem ainda como função, o armazenamento das aulas em uma base dados, para que as mesmas possam ser utilizadas posteriormente pelos próprios autores ou por terceiros.

2.2 CORBA

Dentre as várias plataformas existentes para o desenvolvimento de aplicações distribuídas baseadas em objetos, uma merece destaque. CORBA (*Common Object Request Broker Architecture*) é uma arquitetura proposta por um consórcio internacional de empresas desenvolvedoras de *software*, chamado OMG – *Object Managment Group* [16], cujo objetivo é a criação de um padrão para comunicação entre objetos, independente do *hardware* e *software* para os quais estes tenham sido projetados.

Sua estrutura baseia-se numa arquitetura chamada OMA (*Object Managment Architecture*), que define uma série de elementos para a comunicação de objetos distribuídos. O principal elemento desta arquitetura é um componente chamado ORB – *Object Request Broker*, que funciona como um *middleware*, realizando toda a comunicação entre os objetos distribuídos, levando pedidos de clientes a servidores, retornando os resultados e escondendo destes a localização dos objetos, linguagem de implementação e plataforma de operação.

CORBA tem como forte característica o fato de ser puramente declarativa, onde a ligação entre objetos é feita através de uma linguagem de definição de interface (IDL – *Interface Definition Language*). Esta linguagem define as *interfaces* de cada objeto, onde são determinadas quais as funções que um objeto disponibiliza para seus clientes, como são feitos os pedidos e quais são seus possíveis valores de retorno. A IDL independe da linguagem na qual o objeto foi implementado, tendo sua própria sintaxe derivada do C++, com acréscimos de novas funcionalidades para a especificação de ambientes distribuídos. Através da IDL, CORBA pode gerar mapeamentos para uma série de linguagens, como C, C++, Ada, Smalltalk e Java. A IDL permite que desenvolvedores definam interfaces para seus objetos de forma padronizada, independente de qual linguagem o objeto tenha sido desenvolvido.

Porém a característica que diferencia CORBA de outras tecnologias de objetos distribuídos como JAVA RMI [22] e DCOM [21] são os serviços pré-definidos que a arquitetura especifica, chamados *CORBA services*, considerados de uso geral para todas aplicações que utilizam a especificação. CORBA possui um conjunto de 15 *CORBA services*, incluindo desde um serviço de nomes que serve para localização de objetos até um serviço que permite tornar objetos persistentes. A escolha de CORBA para a especificação e implementação da ambiente de co-autoria proposto foi feita principalmente devido a este serviços oferecidos, sendo que solução considerada como a mais completa, é baseada na utilização de um destes serviços CORBA, o de eventos.

2.3 Modelagem de um sistema de co-autoria

A partir da idéia descrita na seção 2.1, uma solução foi elaborada para que professores possam trabalhar cooperativamente para a produção de cursos e aulas através da Internet. O termo cooperativamente deve ser ressaltado, pois se pretende que nesta solução haja uma efetiva participação de todos professores durante todo o processo de criação de aulas, e não uma simples divisão de tarefas entre os mesmos.

O sistema proposto é baseado na idéia de um ambiente onde potenciais autores de material didático trabalham na construção de uma aula formada por um conjunto de slides, interagindo através da troca de mensagens (*chat* texto no protótipo inicial) e com uma visão compartilhada e única do material didático criado.

Os slides são criados e editados pelos usuários que participam das sessões, porém com a restrição que apenas um usuário pode criar, editar e apagar slides a cada vez. O fato de ter acesso à edição da aula será chamada daqui por diante de possuir o *token* da aula e o usuário que possua o *token* será chamado editor da aula.

O editor da aula terá toda sua ação de navegação entre os slides (ir para o primeiro slide, ir para o próximo slide, etc.) replicada para os demais usuários presentes no ambiente, e toda

edição sobre os slides deverá ser ratificada pelos demais usuários presentes no ambiente através de uma votação majoritária.

Além disso, todo usuário deve ter direito a se tornar o editor da aula. Para garantir tal fato, quando um usuário que não possui o *token* da aula, quiser realizar alguma alteração, inserção ou exclusão de slides, o mesmo deverá requisitar o *token* da aula para si. Com isso, este usuário será inserido numa fila de espera, que determina a ordem de acesso dos usuários à edição da aula. Para garantir que nenhum usuário irá reter o *token* da aula por um tempo indeterminado, todo usuário que assuma o papel de editor da aula terá associado a si um tempo limite para realizar suas edições. Se ao final deste tempo, houver algum usuário na fila de espera pela edição da aula, o *token* é passado automaticamente para o mesmo. Se houver mais de um usuário na fila, a edição é repassada àquele que estiver cadastrado a mais tempo, sendo que quando seu tempo de edição se esgotar, o *token* passará para próximo usuário da fila de acesso e assim por diante. Caso não haja algum usuário requisitando a edição quando o tempo de edição do editor se esgotar, o tempo do atual editor será automaticamente renovado.

Quando um usuário faz uma edição sobre a aula e esta submissão é aceita pelos demais usuários, o mesmo possuirá um direito autoral sobre o conteúdo da aula. Por conta disto, usuários possuem um atributo para diferenciar usuários que já criaram algo ou participaram efetivamente do desenvolvimento da aula daqueles que estão participando da aula pela primeira vez ou não tenham contribuído em nada na criação dos slides. Antes de fazer qualquer submissão o autor é considerado um “autor comum”. O usuário que já realizou alguma modificação ou inserção no material da aula, tendo sido esta edição aceita pelos demais usuários, é considerado então um “autor original”. Tal diferenciação também se faz necessária porque o processo de autoria foi dividido em 2 fases: a fase de criação e a fase de manutenção/visualização da aula.

A fase de criação inicia-se quando não há qualquer slide, e vai até quando todos os “autores originais” decidem terminá-la através de uma votação. Em outras palavras, esta fase inicia-se desde a primeira reunião entre usuários até o momento que os “autores originais” consideram o material criado já em condições de ser apresentado a seus alunos. Para garantir que um único “autor original” não possa encerrar sozinho a fase de criação, a votação que encerra esta fase só poderá ser realizada quando mais da metade dos autores originais da aula estiver presente no ambiente.

Quando encerrada a fase de criação, inicia-se então a fase de manutenção/visualização da aula. Nesta nova fase, edições sobre o conteúdo das aulas ainda poderão ser realizadas, por autores originais ou não, porém com a restrição que um usuário não poderá mais alcançar o status de “autor original”, mesmo que faça uma inserção ou uma alteração de slide(s). Um “autor original” sempre poderá fazer alterações no material criado, independente da presença ou não de outros autores, originais ou não, no ambiente. Um “autor não-original” só poderá editar a aula, caso exista pelo menos um autor original presente no ambiente no mesmo momento em que este autor não-original desejar realizar alguma edição.

Autores originais e não-originais também diferem em como é feito o seu acesso no ambiente. Um “autor original” tem seu acesso sempre garantido, enquanto que um “autor não-original” deverá esperar por uma votação majoritária feita apenas entre os autores originais presentes no ambiente, para que tenha seu acesso permitido. Caso não haja algum “autor original” presente, um “autor não-original” poderá entrar no ambiente, mas nenhum tipo de edição será habilitada. Esta edição só será habilitada quando pelo menos um “autor original” entrar no ambiente.

Cada uma das fases do processo de co-autoria pode ser dividida em vários momentos. Usuários podem se retirar e depois retornar para o ambiente para continuar os debates e a criação das aulas. Desta forma, o grupo de usuários que trabalha na construção de uma aula

pode se reunir em horários específicos e o processo de criação das aulas pode ser realizado em vários dias, por exemplo.

3 Arquiteturas para co-autoria de aulas utilizando CORBA

Criada a idéia geral do ambiente de co-autoria, foram então elaboradas arquiteturas para a implementação da visão descrita na seção 2.3. São apresentadas a seguir, três arquiteturas baseadas no uso de objetos distribuídos CORBA para a representação dos elementos da arquitetura. Todas as arquiteturas são baseadas no uso de três elementos principais que serão descritos a seguir.

3.1 Componentes das arquiteturas

O primeiro componente é o objeto autor, representando cada usuário do ambiente. Em todas as arquiteturas elaboradas suas principais funções são o envio e recebimento de mensagens de outros usuários, assim como do slide que está sendo editado pelo editor da aula e que deve ser apresentado para todos os autores do ambiente. O objeto autor deve também receber e enviar pedidos de voto sobre edição de slides e entrada de novos autores no ambiente.

O segundo componente é chamado servidor de slides, e representa o conjunto de slides criados pelos usuários e que compõe a aula. O nome “servidor de slides” é utilizado para uma melhor compreensão das arquiteturas propostas neste trabalho, pois imagina-se que na realidade deva existir um servidor genérico do qual o servidor de slides faça parte. Este servidor, que poderia ser considerado o servidor de aulas de todo um sistema distribuído de co-autoria, seria formado por servidores de slides, servidores de vídeo, servidores de áudio, dentre outros. As funções deste elemento basicamente refletem a navegação e edição sobre este conjunto, como por exemplo: ir para o próximo slide, ir para o último slide, inserir novo slide, etc. Este elemento deve também fornecer informações sobre a quantidade de slides cadastrados, bem como qual é o slide que está sendo editado pelo editor da aula em um dado instante. De modo a manter persistentes os slides criados durante as sessões de usuários e as relações dos autores originais vinculados a uma aula, o servidor de slides possui uma conexão com um servidor de banco de dados (SGBD), repassando-lhe as respectivas informações criadas no objeto. Desta forma, o objeto pode recuperar os dados de cada aula toda vez que o mesmo for reiniciado.

O último componente básico é chamado de coordenador, sendo o objeto responsável pelo controle da consistência da aula. Este elemento guarda consigo o nome de todos os usuários presentes no ambiente, o editor da aula e a fila dos usuários que estejam requerendo o acesso à edição da aula em um dado instante. O coordenador gerencia também o processo de votação sobre a edição dos slides, emitindo telas de votação para os demais usuários, contabilizando o resultado e realizando a ação do editor no caso de aprovação da mesma pelos usuários. De forma semelhante, o coordenador também gerencia o processo de votação a respeito da entrada de novos usuários no ambiente. Outra importante função realizada pelo coordenador é o tratamento de tolerância à falhas em caso de problemas com os usuários. Quando um autor tem problemas de conexão, o coordenador deve retirar este autor da lista dos presentes. Caso haja problemas com o editor da aula, o coordenador deve ainda repassar o *token* para outro autor. Havendo autores na fila de espera, o *token* é passado de acordo com a ordem da fila. Caso não haja algum autor na fila, o *token* é passado para outro autor aleatoriamente.

Nas seções seguintes, são apresentadas as três arquiteturas propostas, onde cada elemento é apresentado como um objeto distribuído CORBA. Além dos elementos descritos anteriormente, as arquiteturas apresentam também elementos definidos pela arquitetura CORBA, como o serviço de nomes, no qual todo objeto CORBA deve se registrar, de forma a poder ser referenciado por outro objeto. Outro objeto que faz parte de todas as arquiteturas é o *Internet Trader* [13], que representa a implementação de um serviço de *trading* entre objetos

distribuídos. No paradigma de objetos distribuídos, vários objetos podem oferecer o mesmo tipo de serviço, através de replicação de instância da classe deste objeto. O uso de um *trader* faz com que os objetos clientes possam encontrar objetos servidores baseando-se nos serviços que um objeto oferece, e não pelo nome do objeto. Com isso, um objeto cliente pode escolher dentre os possíveis objetos servidores de um determinado serviço, qual melhor satisfaz sua necessidades em um dado instante. Em uma analogia com uma telefônica, o serviço de *trading* funciona como o serviço de páginas amarelas, enquanto o serviço de nomes funcionaria como as páginas brancas[17]. O objeto que oferece um serviço se cadastra no *trader*, e partir deste momento, os demais objetos que necessitam dos serviços oferecidos por este tipo de objeto podem, consultando o *trader*, conseguir a referência a este objeto e utilizar seus serviços. O *Internet Trader* é um serviço específico para encontrar serviços na Internet. Nas figuras que representam as arquiteturas, os objetos são representados por esferas, que fazem requisições através de pedidos a outros objetos via *interfaces* disponibilizadas por cada elemento para seus potenciais clientes. Estas *interfaces* são representadas por símbolos \perp , que são ligados às esferas, para onde convergem os pedidos, representados por setas (\rightarrow). A seguir, veremos mais detalhadamente cada uma das arquiteturas.

3.2 Arquitetura com o coordenador centralizado

A primeira arquitetura idealizada baseia-se na centralização de toda a comunicação e transmissão de informações entre autores e/ou servidor de slides pelo elemento coordenador. Sendo assim, quando por exemplo, um autor deseja enviar uma mensagem no *chat*, esta mensagem é repassada para o coordenador, que distribui a mesma entre os demais autores presentes. Essa mesma idéia é realizada para a votação e a atualização do slide entre os autores. Quando o editor muda o slide a ser apresentado, ele faz a requisição ao coordenador, que por sua vez se encarrega de buscar no servidor de slides a informação desejada e repassá-la ao editor e todos os demais autores. A Figura 2 representa esta arquitetura.

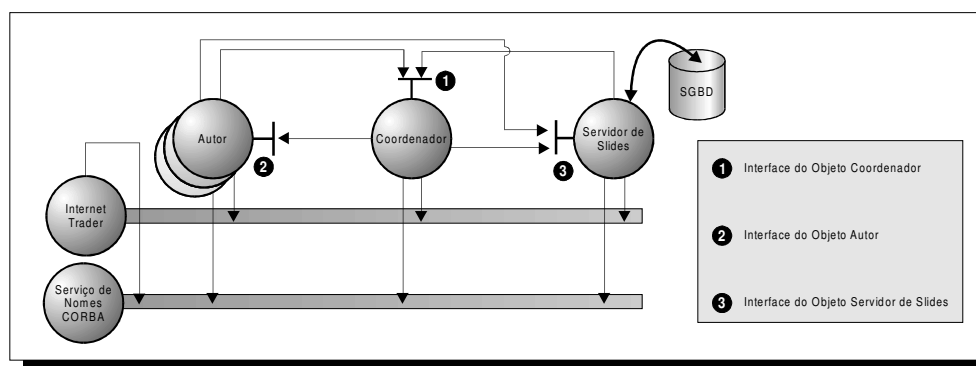


Figura 2 – Arquitetura com centralização do objeto coordenador

Como visto na figura 2, cada objeto componente da arquitetura deve registrar-se no serviço de nomes, para que possa ser referenciado pelos demais objetos. Além disso, os objetos servidor de slides e coordenador devem cadastrar-se também no *Internet Trader* como exportadores de serviço, devido a natureza de suas funções. De forma semelhante, os autores acessam o *Internet Trader*, assumindo papéis de importadores destes serviços.

Esta primeira arquitetura tem como maior vantagem a facilidade de seu desenvolvimento em relação às demais. A centralização de ações pelo coordenador facilita o tratamento de possíveis erros de conexão entre autores, através do isolamento do autor com problemas dos demais. Porém, com isso, há uma sobrecarga de funções por parte do coordenador, que pode causar problemas de desempenho e escalabilidade.

3.3 Arquitetura com o coordenador descentralizado

Esta nova arquitetura é uma variação da arquitetura proposta anteriormente, retirando-se do coordenador o papel de elemento centralizador da comunicação entre autores e o servidor de slides, fazendo com que as comunicações ocorram diretamente entre estes elementos. Para isto, cada autor possui uma lista local, contendo o nome e a referência de outros usuários. Ao entrar no ambiente, esta lista é preenchida com os dados referentes aos autores presentes no ambiente naquele dado instante, notificando também a estes autores a sua presença e fazendo com que cada um destes autores insira o novo autor em sua lista local de autores presentes. A partir deste momento, a comunicação entre os objetos é feita diretamente entre os mesmos. Com isso, por exemplo, quando um autor enviar uma mensagem através do *chat*, o mesmo enviará para cada um de sua lista de autores presentes, uma *string* representando a mensagem. O coordenador continua gerenciando o processo de votação, enquanto o servidor de slides tem as mesmas funções da arquitetura do coordenador centralizado. A Figura 3 descreve esta nova arquitetura.

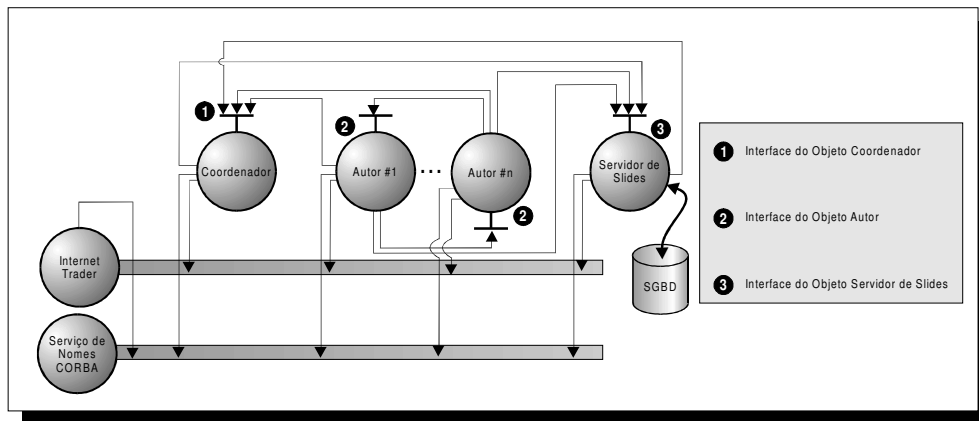


Figura 3 – Arquitetura com descentralização do objeto coordenador

Nesta nova arquitetura, o objetivo de descentralização de tarefas entre os componentes do sistema é alcançado, visto que com a comunicação direta entre autores e entre o servidor de slides, o tráfego de informações pelo coordenador é diminuído sensivelmente. As principais interações entre os autores (troca de mensagem e navegação entre os slides) são feitas sem o intermédio do coordenador, deixando para este elemento, as tarefas de gerenciamento dos processos de votações de edição da aula e de entrada de novos autores, além de também monitorar as possíveis falhas de comunicação com autores. Porém, o ponto negativo desta arquitetura está na complexidade de sua implementação.

3.4 Arquitetura usando o Serviço CORBA de Eventos

A última arquitetura proposta utiliza um dos *CORBA services*, o serviço CORBA de eventos, para fazer a comunicação entre os elementos. Neste serviço, um evento é definido como uma ocorrência em um determinado objeto que é de interesse de outro(s) objeto(s). Quando um evento ocorre, objetos que estão particularmente interessados nesta ocorrência devem ser avisados através de uma notificação enviada pelo objeto que gerou o evento.

Através do serviço de eventos CORBA, objetos também podem enviar dados junto com a própria notificação. Tais dados são representados através da classe *Any* da linguagem de definição de interface (IDL). Esta classe é um tipo especial definido pela IDL CORBA que permite que sejam encapsulados em seu conteúdo, desde tipos básicos (*string*, *integer*, etc.), até tipos complexos, como objetos definidos pelo desenvolvedor, permitindo com isso que qualquer tipo de dado possa ser transmitido através de uma notificação.

Através do serviço de eventos CORBA, objetos assumem dois papéis: produtores (*suppliers*) e consumidores (*consumers*) de eventos. Objetos produtores produzem eventos, enquanto objetos consumidores registram-se para serem notificados quando da ocorrência destes eventos.

O serviço CORBA de eventos utiliza um modelo de comunicação produtor-consumidor que permite que vários objetos produtores de eventos enviem notificações a objetos consumidores através de um chamado *canal de eventos*. Este canal de evento é um objeto CORBA padrão que está situado sobre o ORB, onde acontece a comunicação entre os objetos.

Nesta terceira arquitetura proposta são utilizados quatro canais de eventos para transmissão de notificações e dados entre os elementos da arquitetura, como apresentado na Figura 4.

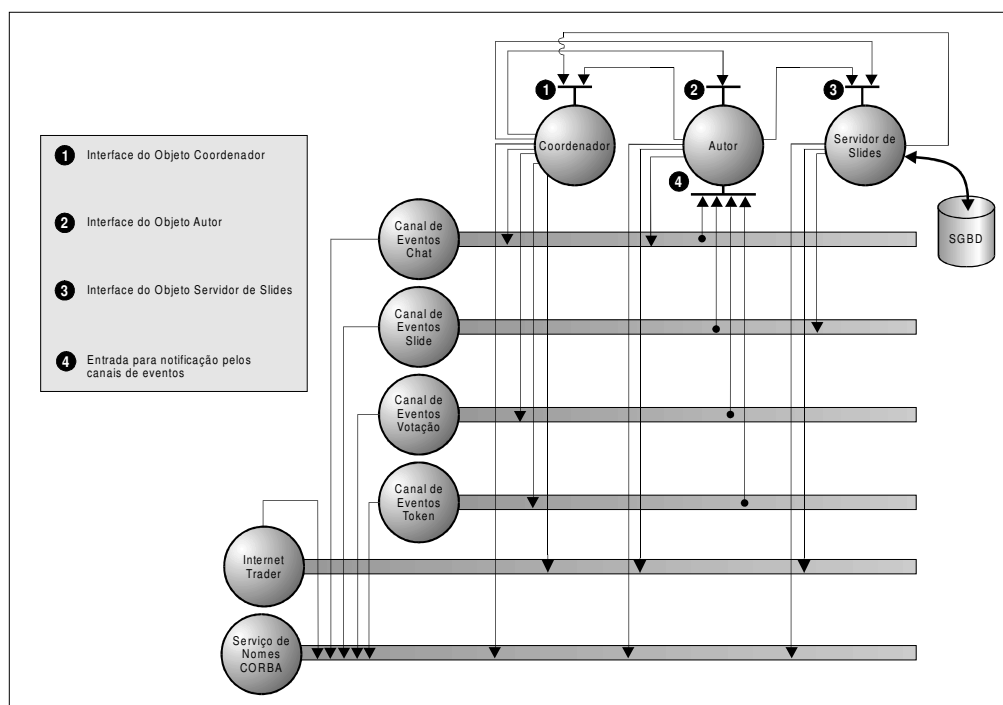


Figura 4 – Arquitetura utilizando serviço de eventos CORBA

De acordo com a figura 4, os quatro canais presentes na arquitetura são utilizados da seguinte forma: um primeiro canal de eventos chamado **chat**, para a comunicação de mensagens; um segundo canal de eventos chamado **slide**, para a transmissão do slide que deve ser apresentado a todos autores; um terceiro canal de eventos chamado **token**, para atualização da lista de usuários presentes e aqueles que estão na lista de acesso ao **token** e por fim, um quarto canal de eventos chamado **votação**, por onde são realizadas as votações sobre edições de slides e entrada de novos usuários.

Nestes canais, os elementos da arquitetura assumem diferentes papéis de produtores e consumidores em cada um deles. No caso do canal de eventos **chat**, todo objeto autor é produtor e consumidor das notificações geradas no canal. Cada notificação leva consigo uma *string*, que representa a mensagem enviada por um autor. Sendo todo autor consumidor deste canal, cada usuário receberá a notificação e consequentemente a mensagem, que será então mostrada numa área específica para o recebimento das mesmas. O coordenador também é um produtor do canal de eventos **chat**, pois este elemento também envia mensagens para serem mostradas como mensagens de **chat**, como por exemplo, a entrada de um novo usuário.

O coordenador é também produtor de eventos para os canais **token** e **votação**. No canal **token**, o coordenador envia periodicamente, a lista de todos os usuários presentes no ambiente. Os

consumidores deste canal são os objetos autor, fazendo com que, todo o autor seja notificado quando da entrada ou saída de um autor do ambiente. No canal votação, o coordenador envia notificações pedindo aos autores seus votos a respeito das ações do editor da aula, quando da realização de uma edição por parte deste usuário. Neste caso, como todo autor é um consumidor deste canal, o editor da aula também será notificado, devendo haver uma filtragem no recebimento da notificação por parte deste usuário, para que o mesmo não seja questionado sobre seu voto. Neste caso, o editor deve ser bloqueado e esperar o resultado da votação, para então continuar sua edição. O último canal, o de slide, serve para que o servidor de slides funcione como o produtor de eventos, repassando para os autores, que são os consumidores do canal, o slide que está sendo editado pelo dono do *token*.

Esta última arquitetura apresentou como principal vantagem a utilização de um serviço pronto, eficaz e fácil de se usar para a comunicação entre os elementos da arquitetura. A forma como o serviço de eventos trata os produtores e consumidores de eventos de cada canal permite que possíveis erros de comunicação entre um único autor e um canal não afetem os demais autores, gerando com isso, uma alta robustez na arquitetura.

4 Prototipagem do ambiente de co-autoria

Durante a fase de especificação das arquiteturas, foi implementado em paralelo, um protótipo desenvolvido em Java e CORBA para o ambiente proposto. A escolha de Java como linguagem para implementação do protótipo foi motivada por tratar-se de uma linguagem orientada a objetos, de fácil aprendizado, robusta, e por possuir bastante documentação a respeito de sua utilização com CORBA. Na fase inicial de desenvolvimento deste protótipo, a implementação seguiu o modelo descrito na arquitetura do coordenador centralizado (seção 4.3). Depois, migrou-se para a arquitetura do coordenador descentralizado (seção 4.4), sendo que por fim, a sua versão final é inteiramente baseada no uso do serviço de eventos CORBA, descrita na seção 4.5. O protótipo consiste de uma janela dividida em quatro áreas principais, como visto na Figura 5.

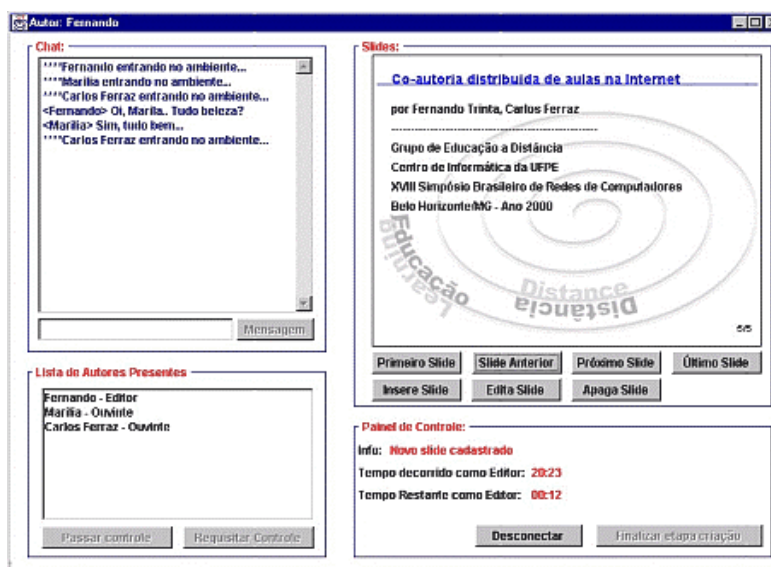


Figura 5 – Tela principal do protótipo.

A área superior esquerda é reservada ao *chat*, por onde toda a comunicação entre os autores é efetuada. Nela, existe um campo onde cada autor pode escrever sua mensagem, e um botão para o envio desta mensagem para os demais autores.

A área inferior esquerda apresenta um quadro onde são relacionados os nomes de todos autores presentes, assim como de seus *status* em relação a aula a cada instante. São três os *status* que os autores podem assumir: quando o autor possui o *token*, seu status é de editor da aula; se o autor está requerendo o acesso à edição da aula, seu status mostra esta informação, junto com a posição que o autor tem na fila de acesso ao *token*; no caso do autor não estar nem editando, nem se candidatando a ter o *token* da aula, o mesmo é considerado um ouvinte. Ainda nesta área, existem dois botões que servem para a mudança do *token* entre os autores. O primeiro botão, com a legenda “requisitar controle”, é utilizado para que autores que desejem acessar o *token* da aula possam se cadastrar na fila de acesso ao mesmo. Para o editor da aula, este botão fica desabilitado. Ao contrário, o segundo botão, com legenda “passar controle”, só é habilitado para o editor da aula e se houver outro autor requisitando o *token* para si. Este botão repassa automaticamente a edição da aula para o autor que esteja requisitando o *token*, e muda o status do antigo editor para ouvinte.

A área superior direita do protótipo é reservada para a visualização da aula, onde é mostrado um slide por vez, assim como botões para navegação sobre a aula, edição ou exclusão do slide que está sendo visualizado ou inserção de novos slides. Estes botões são habilitados apenas para o editor da aula. Nesta área, uma figura funciona como *background* para o slide, sendo mostrado também o número de slides existentes na aula, assim como a ordem do slide atual. Na Figura 5, a notação (5/5) indica que o editor posicionou a aula no quinto slide de um total de cinco existentes.

Por fim, na área inferior direita, existe um espaço reservado para o recebimento de informações de controle do ambiente para o usuário. Primeiro, há a possibilidade de se receber mensagens do sistema, como qual foi a última ação do autor. Como exemplo, na figura 5 é apresentada a mensagem “Novo slide cadastrado”. As demais informações são dados temporais para o atual editor da aula. A primeira informação indica a quanto tempo o atual dono do *token* está editando a aula. A outra informação funciona como um cronômetro regressivo e representa o tempo restante que o dono do *token* tem para realizar suas edições sobre a aula. Quando o tempo do cronômetro se esgota, é verificada a presença ou não de outros autores requisitando o *token*. Caso não exista algum autor na fila, o tempo do editor atual é então renovado. Caso contrário, o *token* é repassado automaticamente para o autor a mais tempo na fila de espera de acesso. Por fim, ainda nesta área, existe um botão com a legenda “Finalizar etapa de criação”, cuja finalidade é realizar uma votação apenas entre os autores originais presentes na aula para o término da fase de criação.

Além desta janela principal, o protótipo é composto de outras janelas secundárias por onde são realizadas a edição de slides e a votação sobre as ações do editor. Quando um editor faz alguma mudança ou inserção de slides na aula, são geradas telas de votação para os demais autores presentes, como descrito na figura 6(a). Cada janela de voto oferece a cada autor três opções de voto: a favor, contra ou abstenção. Estas janelas são mostradas para cada autor por um período máximo de 20 segundos, onde caso o autor não escolha quaisquer das opções, é emitido automaticamente um voto em branco (abstenção) para que a votação não fique indefinidamente a espera do voto deste autor. Durante o período que a votação está sendo realizada, o editor da aula fica bloqueado a espera dos votos, como descrito na figura 6(b). Este janela só desaparece quando do término da votação pelos demais autores.

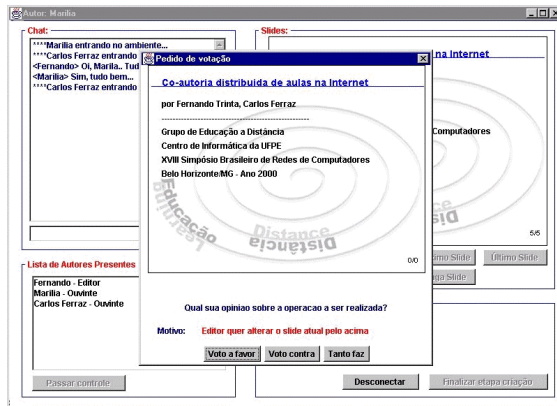


Figura 6(a) – Janela de Votação de Autor sem token

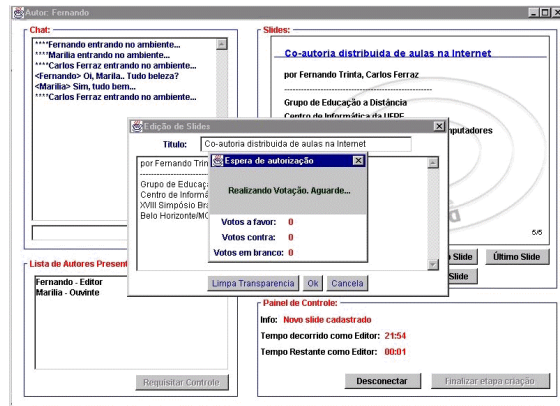


Figura 6(b) – Janela de Bloqueio do Autor com token

4.1 Testes realizados com o protótipo

Como experimento, foram realizados testes com o protótipo na rede local do Centro de Informática e na Rede Metropolitana de Alta Velocidade do Recife, a RecifeATM [28]. Em ambas situações procurou-se explorar ao máximo, diferentes configurações sobre as localizações dos objetos em diferentes plataformas utilizadas, como Windows 95/98/NT/2000, Unix e Linux. Como exemplo, é apresentado abaixo, uma descrição da distribuição dos objetos na rede, durante a execução do protótipo, durante um dos testes.

```
osfind: Found one agent at port 14000
HOST: paulista0
osfind: There are no OADs running on in your domain.
osfind: There are no Object Implementations registered with OADs.
osfind: Following are the list of implementations started manually.
HOST: c5c10
    REPOSITORY ID: IDL:omg.org/CosEventChannelAdmin/EventChannel:1.0
    OBJECT NAME: Chat
    REPOSITORY ID: IDL:comunicacao/Autor:1.0
    OBJECT NAME: Carlos Ferraz
    REPOSITORY ID: IDL:slides/BancoSlides:1.0
    OBJECT NAME: Bancodeslides
HOST: c5c09
    REPOSITORY ID: IDL:comunicacao/Autor:1.0
    OBJECT NAME: Fernando Trinta
    REPOSITORY ID: IDL:comunicacao/Coordenador:1.0
    OBJECT NAME: Coordenador
HOST: c5c11
    REPOSITORY ID: IDL:omg.org/CosEventChannelAdmin/EventChannel:1.0
    OBJECT NAME: Token
    REPOSITORY ID: IDL:comunicacao/Autor:1.0
    OBJECT NAME: Fabiola Pantoja
HOST: paulista
    REPOSITORY ID: IDL:omg.org/CosNaming/NamingContext:1.0
    OBJECT NAME: Autoria/1
    REPOSITORY ID: IDL:omg.org/CosEventChannelAdmin/EventChannel:1.0
    OBJECT NAME: Votacao
    OBJECT NAME: Slide
    REPOSITORY ID: IDL:omg.org/CosNaming/NamingContextFactory:1.0
    OBJECT NAME: Autoria
    REPOSITORY ID: IDL:omg.org/CosNaming/ExtendedNamingContextFactory:1.0
    OBJECT NAME: Autoria
```

Os *hosts* C5C09, C5C10, C5C11 representam microcomputadores pessoais utilizando plataforma Windows NT 4.0, enquanto o *host* paulista é uma Sun UltraSparc Ultra 1 Model 170, com 320 MB de memória, utilizando sistema operacional Solaris. No exemplo acima, os objetos responsáveis pelo funcionamento do serviço de nomes e pelos canais de eventos “Votacao” e “Slide” foram executados no *host* paulista.di.ufpe.br. O canal de eventos

“Chat”, o servidor de slides e o objeto que identifica o usuário Carlos Ferraz foram executados no *host* c5c10; o coordenador foi executado no *host* c5c09, juntamente com o objeto do usuário Fernando Trinta, e por fim, o canal de eventos “Token” e o objeto vinculado ao usuário Fabiola Pantoja foram executados no *host* c5c11.

Nas várias configurações utilizadas durante os testes, o desempenho do protótipo mostrou-se satisfatório, sem grandes atrasos nos envios de mensagem, replicação de slides, entre outros. Em relação aos testes realizados, um aspecto que impossibilitou testes do protótipo na Internet propriamente dita, foi a utilização de esquemas de segurança baseados em *firewalls*, como o utilizado pelo Centro de Informática. O uso de *firewalls* dificulta a comunicação entre os componentes de aplicações baseadas em objetos distribuídos na Internet, como no caso deste protótipo. Nestas situações, onde clientes externos a uma certa rede necessitam acessar objetos localizados em *hosts* protegidos por *firewalls*, é necessário o uso de abordagens baseadas em soluções proprietárias, como o Visibroker Gatekeeper [26] , ou o HP CORBAGate, relatado em [25] . A OMG já busca soluções através de uma especificação sobre uso de aplicações CORBA mesmo quando da presença de *firewalls* [27] , e que possibilite a interação entre objetos situados em diferentes redes.

5 Conclusões

Concluindo este trabalho, vemos a importância da utilização da Internet para fins de educação a distância, através da gama de novas possibilidades educacionais que a rede traz para a EAD. Uma dessas novas possibilidades é a entre grupos de autores de aulas e cursos através da Internet, ressaltando que tal possibilidade ainda não é bem utilizada pelos atuais ambientes que utilizam a Internet para Educação a Distância.

No presente artigo são apresentadas uma visão de co-autoria de cursos por autores de material didático através da Internet e uma modelagem desta visão através de objetos distribuídos, em particular, utilizando a especificação CORBA. São descritas três arquiteturas que possibilitaram a implementação de um ambiente de cooperação para criação de aulas através da Internet, resultando no desenvolvimento de um protótipo utilizando tecnologias como Java e CORBA. Tal protótipo foi testado no Centro de Informática da Universidade Federal de Pernambuco e na rede metropolitana de alta velocidade RecifeATM, e mais testes em sua utilização via Internet estão em andamento. Como trabalhos futuros, o aperfeiçoamento da ambiente visa a inclusão do uso de mídias contínuas a serem apresentadas em conjunto com os slides já criados, novas funções para a apresentação e edição da aula por parte de alunos (fase de manutenção/visualização descrita na seção 2.3), assim como uso de outros serviços CORBA, como o de sincronização, persistência e concorrência entre objetos.

6 Referências

- [1] Aulanet. URL: ead.les.inf.puc-rio.br/aulanet/.
- [2] AUSSERHOFER, Andreas. **Web-Based Teaching and Learning: A Panacea?** IEEE Communications Magazine. Março 1999. p. 92-96.
- [3] BAKER, et al. **The Web as a Distributed Object Infrastructure**. Maio, 1998. URL: www.bodo.org/May98.html
- [4] BARON, J. **Co-Authorship Via the Web for Distance Learners**. Australian Society for Computers in Learning in Tertiary Education Annual Conference, 1997. URL: www.curtin.edu.au/conference/ascilite97/papers/Baron/Baron.html
- [5] CARVER, Carol. **Building a Virtual Community for a Tele-Learning Environment**. IEEE Communications Magazine. Março 1999. p. 114-118.
- [6] COULOURIS, G., DOLLIMORE, J., KINDBERG, T. **Distributed Systems – Concepts and Design**. 2nd edition. Addison-Wesley, 1996.

- [7] CRESPO, S., FONTOURA, M. F. M. C. e LUCENA, C. J. P.. **Um Modelo Conceitual, Compatível com a Plataforma EDUCOM/IMS para Comparação de Ambientes de Educação na Web**. Anais do Simpósio Brasileiro de Informática na Educação, 1998. (CD-ROM).
- [8] **Educação a Distância no WWW**. URL: <http://penta.ufrgs.br/edu/eduwww.html>
- [9] FERRAZ, C. et. al, **Co-autoria Distribuída de Cursos na Internet**. X Simpósio Brasileiro de Informática na Educação. Anais. p. 256 – 263. Curitiba – PR, 23-25 nov. 1999.
- [10] FERRAZ, C. et. al. **Uma Aplicação Distribuída para Educação a Distância na Web**. X Simpósio Brasileiro de Informática na Educação. Anais. p. 248 – 255. Curitiba – PR, 23-25 nov. 1999.
- [11] GINSBERG, A., HODGE, P., LINDSTROM, T., SAMPIERI, B., SHIAU, D.. **"The Little Web Schoolhouse" Using Virtual Rooms to Create a Multimedia Distance Learning Environment**. ACM Multimedia, 1998. URL: www.acm.org/sigmm/MM98/electronic_proceedings/ginsberg/index.html
- [12] HARRIS, Dale A., **Online Distance Education in the United States**. IEEE Communications Magazine. Março 1999. p. 87-91.
- [13] LAWHEAD, P. et al. **The Web and distance learning. What is Appropriate and What is not**. The supplemental proceedings of the conference on Integrating technology into computer science education: working group reports and supplemental proceedings. 1997, p. 27 - 37.
- [14] MACÊDO, R.C., **Internet Trader: Um Serviço para localização de objetos em um ambiente distribuído**. Plano de dissertação de mestrado. URL: <http://www.di.ufpe.br/~rcdm/trader/>
- [15] NUNES, C. C. R., RIBEIRO, J. N. G., FERRAZ, C. A. G. **Investigando Educação a Distância e o Projeto Virtus na UFPE**. Anais do IX Simpósio Brasileiro de Informática na Educação, 1998. (CD-ROM).
- [16] **Object Managment Group Home Page**. URL: www.omg.org
- [17] ORFALI et al. **Instant CORBA**. EUA. Wiley & sons. 1997.
- [18] ORFALI, R., HARKEY, D. **Client / Server Programming with Java and CORBA**. John Wiley & Sons, Inc., 1998.
- [19] **Projeto Virtus – Educação e Ciberespaço**. URL: www.cac.ufpe.br, 1998.
- [20] SANTORO, Flávia et al. **Um framework de Ambientes de Suporte à Aprendizagem Cooperativa**. Anais do IX Simpósio Brasileiro de Informática na Educação, 1998. (CDROM).
- [21] **The Distributed Component Object Model**. URL: <http://www.microsoft.com>.
- [22] **The source for Java™ Tecnology**. URL: <http://java.sun.com>.
- [23] **Visibroker – Naming and Events Services, Programer's guide. Version 3.0**. Visigenic Software, Inc. 1997.
- [24] TRINTA, Fernando A. M., **Arquiteturas distribuídas para co-autoria de aulas na Internet**. Dissertação de Mestrado. Universidade Federal de Pernambuco. Centro de Informática. Abril/2000. 128 p.
- [25] INGHAM, D., REES, O., NORMAN A.: **CORBA Transactions Through Firewalls**. Proceedings of the International Symposium on Distributed Objects and Applications (DOA'99), 1999. 21 p.
- [26] **Borland Gatekeeper**. URL: <http://www.borland.com/gatekeeper>
- [27] **Joint Revised Submission CORBA/Firewall Security**. Maio, 98. Online. Capturado na Internet. URL: <http://www.omg.org/pub/docs/orbos/98-07-04.pdf>
- [28] **Rede RecifeATM**. URL: <http://www.cin.ufpe.br/~recifeatm>